

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

A. KOMPETENSI

1. Memahami pengertian dan manfaat fungsi
2. Memahami fungsi tanpa nilai balik
3. Memahami fungsi dengan nilai balik
4. Memahami jenis parameter: masukan, keluaran, dan masukan/keluaran
5. Memahami parameter Pass by value & pass by reference
6. Memahami parameter bertipe array, konstan, dan parameter dengan nilai default

B. ALAT DAN BAHAN

1. PC/ Laptop
2. Flowchart application
3. C++ (atau bahasa pemrograman lain yang dikuasai)
4. Labsheet

C. KAJIAN TEORI

Fungsi merupakan kumpulan statemen yang dikelompokkan menjadi satu bagian kode (blok diagram) untuk menyelesaikan tugas spesifik tertentu. Melalui cara yang demikian maka kode (fungsi) hanya perlu didefinisikan sekali akan tetapi dapat digunakan berulang kali tanpa harus menuliskan kode yang sama.

Pada bahasa pemrograman C++, fungsi dibedakan menjadi 2, yaitu "user defined function", yaitu fungsi yang didefinisikan sendiri dan "built in function", yaitu fungsi siap pakai yang telah disediakan oleh compiler.

Program merupakan kumpulan dari fungsi, baik yang didefinisikan secara langsung dalam program maupun yang dideklarasikan dalam suatu file header. Pada C++, fungsi utama disebut fungsi main (). Fungsi main () akan dipanggil pertama kali ketika program dijalankan.

1. Mendefinisikan fungsi

Bentuk umum:

```
Tipe_kembalian nama_fungsi(daftar_parameter) {
//badan fungsi
}
```

Keterangan:

Tipe_kembalian: fungsi bisa saja mempunyai nilai balik. Tipe kembalian adalah tipe data dari nilai yang akan dikembalikan oleh fungsi

Nama_fungsi: adalah nama yang akan digunakan dalam pemanggilan. Pemanggilan fungsi harus disesuaikan dengan daftar parameter yang disertakan pada saat pendefinisian fungsi.

Daftar_parameter: dalam suatu fungsi bisa memiliki lebih dari satu parameter. Parameter merupakan variabel berisi nilai yang akan disertakan ke dalam proses yang dilakukan oleh fungsi.

2. Fungsi tanpa nilai balik

Fungsi yang tidak memiliki nilai balik dinyatakan dengan fungsi tipe void. Bentuk umum:

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

```
Void nama_fungsi (parameter1, parameter2, ....) {
    Statemen_yang_akan_dilakukan;
    ....
}
```

Contoh1:

Fungsi yang menuliskan “saya pasti mahir C++” sebanyak 10 kali.

```
#include <iostream>

using namespace std;

// Membuat fungsi dengan nama Tulis10Kali()
void Tulis10Kali() {
    for (int C=.....; .....; .....) {
        cout<<"saya pasti mahir C++"<<endl;
    }
}

// Fungsi utama dalam program C++
int main() {

    // Memanggil fungsi Tulis10Kali()
    // untuk dieksekusi
    Tulis10Kali();

    return 0;
}
```

Note: isilah titik-titik di tempat yang bertanda kuning.

Hasil yang diharapkan:

```
saya pasti mahir C++
```

3. Fungsi dengan nilai balik

Bentuk umum:

```
Tipe_data nama_fungsi(parameter1, parameter2, ...) {
    Statemen_yang_akan_dilakukan;
    .....
    return nilai_yang_akan_dikembalikan;
}
```

Fungsi dengan nilai balik sebenarnya merupakan sebuah nilai atau ekspresi, oleh karena itu pemanggilannya dapat dilakukan seperti pemanggilan sebuah variabel. Sebagai contoh, nilai yang dikembalikan dapat ditampung ke dalam suatu variabel tertentu yang memiliki tipe data sama dengan tipe kembalian fungsi bersangkutan.

```
nama_variabel=nama_fungsi(parameter1, parameter2, ...);
```

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

pemanggilan fungsi juga dapat dilakukan langsung melalui perintah cout, jika yang diperlukan adalah ingin menampilkan nilai yang dihasilkan oleh fungsi.

```
Cout<<nama_fungsi(parameter1, parameter2, ...);
```

4. Fungsi yang mengembalikan tipe string

Contoh 2:

```
#include <iostream>

using namespace std;

// Membuat fungsi sederhana
// yang mengembalikan tipe string
char* TestFungsiString() {
    return (char*) "Ini adalah nilai dari fungsi";
}

// Fungsi utama
int main() {
    // Memanggil dan menampilkan hasil fungsi
    cout<<TestFungsiString();

    return 0;
}
```

5. Fungsi yang mengembalikan tipe bilangan

Contoh 3:

```
#include <iostream>

using namespace std;

// Membuat fungsi dengan nilai kembalian
// bertipe double
double TestFungsiBilangan() {
    return (3.14 * 2);
}

// Fungsi utama
int main() {

    cout<<"Nilai yang terdapat dalam fungsi: ";
    cout<<TestFungsiBilangan();

    return 0;
}
```

6. Fungsi dengan parameter

Fungsi dengan parameter memberikan hasil yang bersifat dinamis tergantung dari nilai parameter yang dimasukkan.

7. Jenis parameter

a. Parameter masukan

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

Parameter ini akan digunakan sebagai nilai masukan dalam sebuah fungsi. Nilai tersebut kemudian akan diproses oleh fungsi untuk menghasilkan sebuah nilai kembalian (return value).

Contoh 4:

```
#include <iostream>

using namespace std;

// Membuat fungsi dengan parameter
// bertipe masukan
int TambahSatu(int X) {
    int hasil;
    hasil = X + 1;
    return hasil;
}

int main() {

    /* Mendeklarasikan variabel
       yang akan digunakan sebagai
       nilai parameter pada saat pemanggilan */
    int Bilangan, HASIL;

    cout<<"Masukkan sebuah bilangan bulat: ";
    cin>>Bilangan;

    HASIL = TambahSatu(Bilangan);

    // Menampilkan nilai
    // setelah diproses di dalam fungsi
    cout<<"Nilai akhir: "<<HASIL;

    return 0;
}
```

Note:

Variabel x sebagai parameter masukan dan $var\ x$ disebut sebagai parameter formal, sedangkan Bilangan disebut parameter aktual.

b. Parameter keluaran

Parameter keluaran merupakan parameter yang berfungsi untuk menampung nilai yang akan dikembalikan. Parameter tersebut berguna sebagai nilai keluaran (output) dari sebuah fungsi. *Parameter keluaran harus berupa pointer maupun referensi* dan pada umumnya terdapat pada fungsi yang tidak mengembalikan nilai.

Contoh 5:

```
#include <iostream>

using namespace std;

// Parameter X sebagai masukan
// dan parameter hasil sebagai keluaran
void TambahSatu(int X, int *hasil) {
```

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

```

    *hasil = X + 1;
}

int main() {

    int Bilangan, HASIL;

    cout<<"Masukkan sebuah bilangan bulat: ";
    cin>>Bilangan;

    // Memanggil fungsi TambahSatu()
    TambahSatu(Bilangan, &HASIL);

    // Menampilkan nilai
    // setelah diproses di dalam fungsi
    cout<<"Nilai akhir: "<<HASIL;

    return 0;
}

```

c. Parameter masukan/keluaran

Parameter ini digunakan sebagai masukan dan juga keluaran.

Contoh 6:

```

#include <iostream>

using namespace std;

// Parameter X sebagai masukan
// dan juga sebagai keluaran
void TambahSatu(int *X) {
    *X= *X + 1;
}

// Fungsi utama
int main() {

    int Bilangan;

    cout<<"Masukkan sebuah bilangan bulat: ";
    cin>>Bilangan;

    // Memanggil fungsi TambahSatu()
    TambahSatu(&Bilangan);

    // Menampilkan nilai
    // setelah diproses di dalam fungsi
    cout<<"Nilai akhir: "<<Bilangan;

    return 0;
}

```

8. Melewatkan parameter: pass by value dan pass by reference

Terdapat dua metode yang digunakan untuk melewatkan parameter ke dalam sebuah fungsi, yaitu:

a. Berdasarkan nilai (*pass by value*)

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

Fungsi ini akan melewati nilai parameter ke dalam sebuah fungsi untuk digunakan sesuai proses yang terdapat di dalam fungsi tersebut. Jika parameter dilewatkan dengan cara ini maka nilai yang dihasilkan oleh fungsi tersebut tidak akan mempengaruhi nilai yang terdapat pada program (di luar fungsi tersebut). Hal ini dikarenakan pada saat pemanggilan fungsi, compiler hanya akan membuat salinan (copy) dari nilai yang terdapat pada parameter aktual ke parameter formal. Atau dengan kata lain yang akan berubah adalah nilai di dalam fungsi saja.

Contoh 7:

```
#include <iostream>

using namespace std;

// Membuat fungsi dengan melewati
// nilai X ke dalamnya
void Kali2(int X) {
    X = X * 2;
    cout<<"Nilai di dalam fungsi: "<<X<<endl;
}

// Fungsi utama
int main() {
    int Bilangan;

    cout<<"Masukkan sebuah bilangan bulat: ";
    cin>>Bilangan;
    cout<<endl;

    // Menampilkan nilai awal
    cout<<"Nilai awal: "<<Bilangan<<endl;

    // Memanggil fungsi Kali2
    Kali2(Bilangan);

    // Menampilkan nilai akhir
    cout<<"Nilai akhir: "<<Bilangan<<endl;

    return 0;
}
```

Hasil:

```
Masukkan sebuah bilangan bulat: 2
Nilai awal: 2
Nilai di dalam fungsi: 4
Nilai akhir: 2
```

b. Berdasarkan referensi atau alamatnya (*pass by reference*)

pass by reference adalah metode untuk melewati parameter ke sebuah fungsi berdasarkan alamatnya. Jika membuat alias (referensi) dari sebuah variabel maka alamat dari variabel dan alias tersebut adalah sama. Hal ini merupakan konsep dasar pengiriman parameter berdasarkan alamat.

Berbeda dengan pass by value, melewati parameter dengan cara ini akan menyebabkan nilai yang terdapat di dalam fungsi sama persis dengan nilai yang

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

terdapat pada program (di luar fungsi). Hal ini dikarenakan alamat dari variabel yang berperan sebagai parameter formal sama dengan alamat dari variabel yang berperan sebagai parameter aktual.

Contoh 8:

```
#include <iostream>

using namespace std;

// Mendefinisikan fungsi
// yang melewatkan parameter
// berdasarkan alamat
void Kali2(int& X) {          // Gunakan tanda &
                            // untuk membuat alias
                            // atau referensi

    X = X * 2;
    cout<<"Nilai di dalam fungsi: "<<X<<endl;
}

int main() {
    int Bilangan;

    cout<<"Masukkan sebuah bilangan bulat: ";
    cin>>Bilangan;
    cout<<endl;

    // Menampilkan nilai awal
    cout<<"Nilai awal: "<<Bilangan<<endl;

    // Memanggil fungsi Kali2()
    Kali2(Bilangan);

    // Menampilkan nilai akhir
    cout<<"Nilai akhir: "<<Bilangan<<endl;

    return 0;
}
```

Hasil:

```
Masukkan sebuah bilangan bulat: 5
Nilai awal: 5
Nilai di dalam fungsi: 10
Nilai akhir: 10
```

Nilai di dalam fungsi dan di akhir sama karena x bukanlah sebuah variabel melainkan alias (nama lain) atau referensi dari variabel Bilangan. Alias x dan variabel Bilangan mempunyai alamat yang sama karena keduanya sebenarnya adalah objek yang sama. Oleh karena itu, bila nilai x berubah maka nilai yang terdapat pada variabel Bilangan pun juga akan berubah.

9. Parameter bertipe array

Array dapat pula digunakan sebagai parameter dalam sebuah fungsi.

Contoh 9:

```
#include <iostream>
```

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

```

using namespace std;

// Mendefinisikan fungsi
// untuk proses input array
void InputArray(int A[], int N) {
    for (int C=0; C<N; C++) {
        cout<<"Masukkan nilai A["<<C<<"]: ";
        cin>>A[C];
    }
}

// Mendefinisikan fungsi
// untuk menghitung jumlah (sum)
// dari semua elemen array
long Jumlah(int A[], int N) {
    long jml = 0;
    // Menjumlahkan semua elemen array
    for (int C=0; C<N; C++) {
        jml += A[C];
    }
    return jml;
}

// Fungsi utama
int main() {
    int X[100]; // Elemen maksimal adalah 100
    int BanyakElemen;
    long HASIL;

    cout<<"Masukkan banyaknya elemen "
        <<"yang diinginkan: ";
    cin>>BanyakElemen;
    cout<<endl;

    // Memanggil fungsi InputArray
    InputArray(X, BanyakElemen);

    // Memanggil fungsi Jumlah()
    // dan menampung hasilnya
    // ke variabel HASIL
    HASIL = Jumlah(X, BanyakElemen);

    // Menampilkasn hasil
    cout<<"\nJumlah elemen array = "
        <<HASIL;

    return 0;
}

```

10. Parameter konstan

Parameter sebuah fungsi dapat bersifat tetap (konstan), yang berarti di dalam fungsi nilai parameter tidak dapat dirubah. Untuk melakukan ini hanya perlu menambahkan kata kunci `const` di depan deklarasi parameter tersebut.

Contoh 10:

```

#include <iostream>

using namespace std;

```

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

```

// Mendefinisikan fungsi
// untuk menghitung keliling lingkaran
double KelilingLingkaran(const float PI,
                          int jari_jari) {
    return (2*PI*jari_jari);
}

// Fungsi utama
int main() {
    int r;
    double HASIL;

    cout<<"Masukkan panjang jari-jari lingkaran: ";
    cin>>r;

    // Memanggil fungsi KelilingLingkaran
    HASIL = KelilingLingkaran(3.14, r);

    // Menampilkan hasil yang didapatkan
    cout<<"Keliling lingkaran = "<<HASIL;

    return 0;
}

```

11. Rekursi

Rekursi adalah fungsi yang pada saat pendefinisianya memanggil dirinya sendiri untuk melakukan proses di dalamnya.

Contoh:

```

#include <iostream>

using namespace std;

// Mendefinisikan fungsi Faktorial()
int Faktorial(int X) {
    if (X==1) return(1);
    // Memanggil dirinya sendiri
    return X * Faktorial(X-1);
}

// Fungsi utama
int main() {
    int Bilangan, HASIL;
    cout<<"Masukkan bilangan yang akan dihitung: ";
    cin>>Bilangan;

    // Memanggil fungsi Faktorial()
    HASIL = Faktorial(Bilangan);

    // Menampilkan hasil
    cout<<Bilangan<<"! = "<<HASIL;

    return 0;
}

```

D. LANGKAH KERJA

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	FUNGSI	
	No. LST/TE/EKA5208/08	Revisi : 01	Tgl : 20 Februari 2017

1. Berdoalah sebelum memulai pekerjaan.
2. Baca dan pahami labsheet yang diberikan.
3. Kerjakan contoh-contoh hasil program di atas .
4. Apabila mengalami kesulitan atau pekerjaan telah selesai, konsultasikan pada dosen pengampu.
5. unggah hasil pekerjaan anda di bestmart pada mata kuliah algoritma dan struktur data.

E. TUGAS

1. Pelajari dan pahami setiap kode program di atas.
2. Buatlah algoritma dengan bahasa narasi, flowchart, pseudocode, dan program di atas.
3. Buatlah 3 algoritma dan program dengan fungsi di dalamnya.

F. DAFTAR PUSTAKA

Kajian pustaka diambil dari buku:

Budi Raharjo. 2015. Pemrograman C++ Mudah dan Cepat Menjadi Master C++. Bandung: Penerbit Informatika.